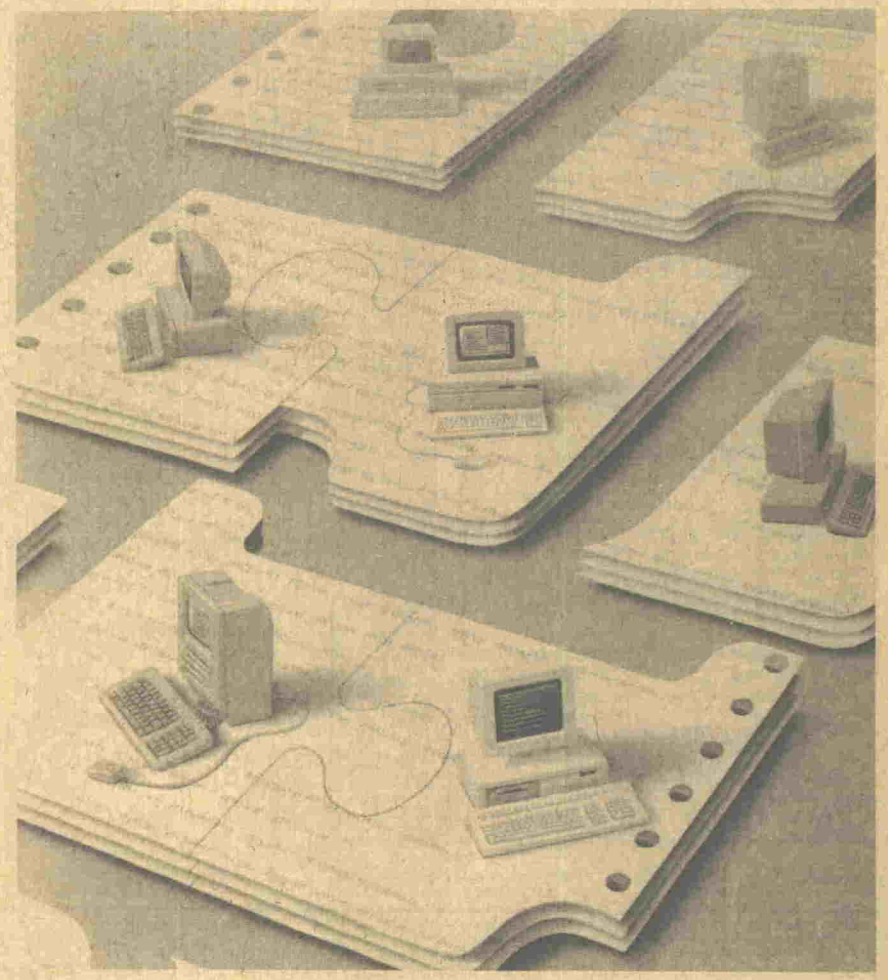
อาจหาญ สัตยารักษ์

SOFT MAIL

ทางเลือกใหม่ในการจัดการฐานข้อมูลดีเบส (1)

ใช่ว่าจะมีวิธีจัดการฐานข้อมูลด้วยดีเบสเท่านั้น เราสามารถเขียนโปรแกรมด้วยภาษาปาสคาลเองก็ได้



เมื่อเรามีความประสงค์จะดำเนินการกับข้อมูลชุดหนึ่ง ๆ นั้น ถ้าเป็นการใช้เครื่องคอมพิวเตอร์เรามักจะนึกถึงโปรแกรมสำเร็จรูปฐานข้อมูลดีเบสเป็นอันดับแรก เพราะเป็นโปรแกรมที่มีระบบจัดการเกี่ยวกับฐานข้อมูลครบถ้วนในทุกเรื่อง การจัดการฐานข้อมูลด้วยวิธีอื่น เช่น เขียนโปรแกรมขึ้นมาเองทั้งหมดนั้นต้องใช้เวลาในการพัฒนาโปรแกรม คนส่วนมากจึงรู้ดีว่าดีเบสจะช่วยดำเนินการในเรื่องนี้ได้

ในฐานะที่ผู้เขียนชอบเขียนโปรแกรมภาษามีความคิดเห็นว่า เราน่าจะมีทางเลือกที่ดีกว่าคือ ใช้ระบบเก็บฐานข้อมูลด้วยดีเบส ผู้ที่ใช้ดีเบสเก่งก็ทำงานด้วยดีเบสไป ผู้เขียนไม่อยากใช้ดีเบสโดยตรงก็สามารถเขียนโปรแกรมภาษาจัดการกับฐานข้อมูลดีเบสได้เช่นกัน โดยการจัดการฐานข้อมูลด้วยภาษาจะยังคงไว้ซึ่งส่วนของโครงสร้างเดิมของดีเบสไม่มีการเปลี่ยนแปลง ผลก็คือ คนเขียนโปรแกรมภาษาได้เปรียบกว่าในกรณีที่สามารถใช้โปรแกรมภาษาจัดการกับฐานข้อมูลดีเบสได้ แต่คนใช้ดีเบสไม่สามารถใช้ดีเบสจัดการกับฐานข้อมูลที่พัฒนาขึ้นด้วยโปรแกรมภาษาจริง ๆ (คนละประเด็นกับที่จะอธิบายต่อไป)

ผู้เขียนขอทำความตกลงกันก่อนว่า การจัดการกับฐานข้อมูลถ้าผู้อ่านประสงค์จะใช้ภาษาไทยก็ต้องจัดการกับระบบภาษาไทยของการ์ดแสดงผลจอภาพที่ใช้อยู่เป็นอันดับแรก เพราะโปรแกรมที่จะพูดถึงต่อไปนี้รันตามระบบที่ผู้อ่านใช้งานได้อย่างแน่นอน ส่วนโปรแกรมดีเบสที่ใช้นี้คือดีเบสทรี และบทความชุดนี้ได้จัดเป็นซอฟต์เมล์เช่นเดิมโดยเก็บไว้ในไฟล์ชื่อ dbasel.pas ในตอนที่ 1 จะศึกษาถึงโครงสร้างฐานข้อมูล การแสดงข้อมูล และการเพิ่มข้อมูล

ระบบจัดการฐานข้อมูลดีเบส

ก่อนที่จะเขียนโปรแกรมภาษาเพื่อจัดการกับฐานข้อมูลใด ๆ เราจะต้องศึกษาโครงสร้างของฐานข้อมูลนั้น ๆ ก่อน สมมติผู้เขียนใช้ดีเบสสร้างฐานข้อมูลชื่อ test.dbf เป็นตัวอย่าง สำหรับแฟ้มข้อมูลที่ให้มาในซอฟต์เมล์นั้นมีระเบียนข้อมูล (record) เพียง 3 ระเบียน ข้อมูลเท่านั้นที่จะทำการเพิ่มและแก้ไขข้อมูลด้วยโปรแกรมภาษาปาสคาล

เมื่อเราได้ทำการตรวจสอบโครงสร้างด้วยคำสั่ง list stru เราจะเห็นโครงสร้างฐานข้อมูลของแฟ้มข้อมูลชื่อ test.dbf ดังรูปที่ 1

ส่วนรูปที่ 2 เป็นโครงสร้างแฟ้มข้อมูลที่เก็บไว้ในแผ่นบันทึก ผู้เขียนลอกมาจากหน้าจอภาพด้วยการกดแป้น PrintScreen หลังจากใช้ไฟล์ SM.COM (ซึ่งมีอยู่ในซอฟต์เมล์ด้วยแล้ว) เรียกมาแสดงผลบนจอภาพ

จากรูปที่ 1 จะพบส่วนที่ควรให้ความสนใจดังนี้

1. วันที่ที่ทำการแก้ไขแฟ้มข้อมูลครั้งสุดท้าย

2. จำนวนระเบียนข้อมูลซึ่งมีทั้งหมด 430 ระเบียนข้อมูล

3. ความยาวของหนึ่งระเบียนข้อมูล ตามรูปมีอยู่ 89 ไบต์

4. ชื่อของเขตข้อมูล (field) ชนิดของเขตข้อมูล ความกว้างของเขตข้อมูล (ถ้าสังเกตให้ดีจะพบว่าความยาวของหนึ่งระเบียนข้อมูลจะมากกว่าผลรวมของความกว้างของเขตข้อมูลทั้งหมดอยู่ 1 ไบต์)

เราศึกษาโครงสร้างการเก็บสิ่งต่าง ๆ เหล่านี้ได้จากรูปที่ 2 (บางส่วนของรูปอาจไม่ตรงกับความจริงของแฟ้มข้อมูล เพราะผู้เขียนได้แก้ไขเพื่อช่วยให้การพิมพ์ไม่เปลี่ยนตามรหัสแอสกีบางตัว) ดังนี้

1. วันที่ที่แก้ไขแฟ้มข้อมูลครั้งสุดท้ายถูกเก็บไว้ที่ออฟเซต 1-3 (นับ 0 ที่ตำแหน่งเริ่มต้น) ดังนี้

– ออฟเซตที่ 1 เก็บ ค.ศ.ตามรูปเป็น 5A ซึ่งเป็นเลขฐาน 16 เราเปลี่ยนเป็นเลขฐาน 10 ได้ผลลัพธ์เท่ากับ 5A (ฐาน 16) = 01011010 (ฐาน 2) = 64 + 16 + 8 + 2 = 90 (ฐาน 10)

– ออฟเซตที่ 2 เก็บเลขลำดับของเดือน ตามรูปเก็บ 0B คือ 0+8+2 + 1 = 11

– ออฟเซตที่ 3 เก็บวันที่ ตามรูป

```
. list stru
Structure for database : C:test.dbf
Number of data records :      430
Date of last update    : 11/09/90
Field  Field name  Type        Width    Dec
    1  NUMBER      Numeric         3
    2  NAME        Character      16
    3  NAME2       Character      16
    4  UNIT        Character      10
    5  SCHOOL      Character      15
    6  POSITION    Character       7
    7  LEVEL       Character       1
    8  DISTRICT    Character      10
    9  PROVINCE    Character      10
** Total **                      89
```

รูปที่ 1 แสดงโครงสร้างฐานข้อมูลที่อ่านได้จากคำสั่ง list stru

```
Side 0, Track 25, Sector 9 the cursor is at offset  0 decimal,   0 hexadecimal
  1st sector of TEST.DBF
035A0B09 AE010000 4Z015900 00000000 00000000 00000000 P¾  AY
00000000 00000000 4E554D42 45520000 0000004E 0900024B          NUMBER     N K
03000000 01000000 00000000 00000000 4E414D45 00000000                NAME
00000043 0000004B 10000000 00000000 00000000 00000000     C   K
4E414D45 32000000 00000043 1C00024B 10000000 01000000 NAME2       C K
00000000 00000000 554E4954 00544D45 4E540043 2C00024B           UNIT TMENT C, K
00000000 01000000 00000000 00000000 5343484F 4F4C0000                 SCHOOL
00000043 3600024B 00000000 00000000 00000000 00000000     C6 K
504F5349 54494F4E 00000043 4500004B 00000000 01000000 POSITION    CE  K
00000000 00000000 4C455645 4C000000 00000043 4C00024B          LEVEL      CL K
01000000 01000000 00000000 00000000 44495354 52494354                DISTRICT
00000043 4D00024B 00000000 01000000 00000000 00000000     CM K
50524F56 494E4345 00000043 5700004B 00000000 01000000 PROVINCE    CW  K
00000000 00000000 00203236 31B72EC7 2EB2CEC1 B7DA2020           261น.ส.ดารน
20202020 20A3C4CE BFD4BCB8 D8C3C0E4 20202020 20C7CEBF       ความไพบูลย์      สาม
DDABC5DB A1C6CEC3 CFB9CEA5 A1DDC3C0 CEB1DA20 2020CA2E ฺฺฺ
```

คือ 09 ซึ่งมีค่าเลขฐานสิบเท่ากับ 9 เช่นกัน จะเห็นว่าตรงกับวันที่ที่ปรากฏในรูปที่ 1

2. จำนวนระเบียนข้อมูลถูกเก็บไว้ที่ออฟเซต 4-5 ตามรูปที่ 2 คือ 4 ตัวแรกของคอลัมน์ที่สองจะเป็น AE-01 (offset 4=AE offset 5=01) ซึ่งเป็นเลขฐาน 16 ลองเปลี่ยนเป็นฐาน 10 จะได้เป็น AE = 128 + 32 + 8 + 4 + 2 = 174 และ 01 = 1 ดังนั้น AE01 = 174 + 1 × 256 = 430 ใช้วิธีให้ไบต์แรกเก็บเศษของ 430/256 และไบต์ที่สองเก็บจำนวนเต็มของ 430/256 ทำให้เก็บระเบียนข้อมูลได้สูงสุด 65535 ระเบียนข้อมูล

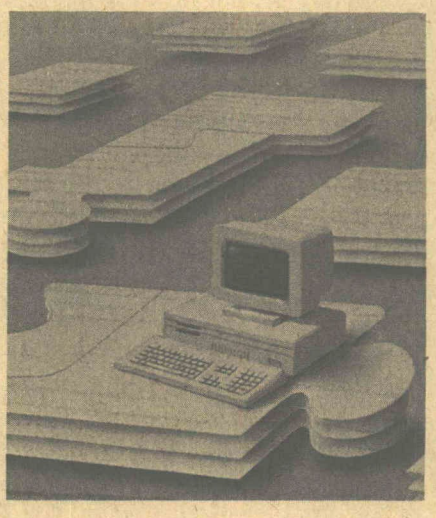
3. ความยาวของระเบียนข้อมูลถูกเก็บที่ออฟเซต 10 และ 11 จะเห็นเป็น 5900 (สี่ตัวหลังของคอลัมน์ที่สาม) เปลี่ยนเป็นเลขฐาน 10 ได้เท่ากับ 59 (ฐาน 16) = 01011001 (ฐาน 2) = 64 + 16 + 8 + 1 = 89 (ฐาน 10)

4. ชื่อเขตข้อมูลที่ 1, 2, 3..... (มีความยาว 10 ไบต์ทุกชื่อเขตข้อมูล) จะเริ่มที่ออฟเซต 32, 64, 96, ...ตามลำดับ กล่าวคือ ใช้เนื้อที่ 32 ไบต์ต่อการเก็บส่วนที่เกี่ยวกับเขตข้อมูล ชนิดของเขตข้อมูลถูกเก็บ ณ ออฟเซตที่นับจากไบต์แรกของส่วนเก็บชื่อเขตข้อมูลไปอีก 11 ไบต์ ความกว้างของเขตข้อมูลเก็บที่ออฟเซตที่นับจากไบต์แรกของส่วนเก็บชื่อเขตข้อมูลไป 16 ไบต์ (ไบต์ถัดจากนี้คือส่วนที่เก็บจำนวนเลขหลังจุดทศนิยมของเขตข้อมูลชนิดตัวเลข) ตัวอย่างเช่น เขตข้อมูลที่ 1 เป็นดังนี้ (ให้หาตารางแสดงรหัสแอสกีมาลองเทียบดูด้วย) ชื่อเขตข้อมูลอยู่ออฟเซตที่ 32-41 ซึ่งตามรูปที่ 2 จะเป็น 4E554D42 45520000 0000 เปลี่ยน

```
PROGRAM ReadDATA_from_dBASE;
uses crt,dos;
var f:file;
    DataDbase:array[0..52000] of char;
    NoR,CountOfRecord,SizeOfRecord,SizeOfHead,EditNo:word;
    Head:array[0..1024] of byte;
    NameOfField:array[1..30] of string[10];
    TypeOfField:array[1..30] of char;
    WidthOfField:array[0..30] of byte;
    FDec:array[1..30] of byte;
    ch:char;  CountOfField,Appendck:byte;
    Dbasefile,DateStr:string;
```

รูปที่ 3 แสดงตัวแปรที่กำหนดไว้ในส่วนหัวของโปรแกรมภาษาปาสคาล

เป็นข้อความได้ดังนี้ 4E=N, 55=U, 4D=M, 42=B, 45=E, 52=R = NUMBER ชนิดของเขตข้อมูลอยู่ออฟเซตที่ 32 + 11=43 ซึ่งจะเก็บ 4E หรือ N (คือ numberic นั่นเอง) ความกว้างของเขตข้อมูลอยู่ที่ออฟเซต 32 + 16=48 ตามรูปเก็บ 03 เขตข้อมูลที่ 2 จะเป็นดังนี้ ชื่อเขตข้อมูลอยู่ออฟเซตที่ 64-73 ซึ่งตามรูปที่ 2 จะเป็น 4E414-D45 00000000 0000 เปลี่ยนเป็นข้อความได้ดังนี้ 4E=N, 41=A, 4D=M, 45=E=NAME ชนิดของเขตข้อมูลในออฟเซตที่ 64 + 11=75 จะเก็บ 43 หรือ C (คือ character) ความกว้างของเขตข้อมูลเป็นออฟเซตที่ 64 + 16=80 ตามรูปเก็บ 10 ในเลขฐาน 16 เปลี่ยนเป็นเลขฐาน 10 ได้ค่าเท่ากับ 16



เขตข้อมูลอื่น ๆ ขอให้ผู้อ่านศึกษาเองแล้วจะเห็นว่า ส่วนแรกของแฟ้มข้อมูลดีเบสนั้นจะเก็บรายละเอียดของโครงสร้างไว้ทั้งหมด ถึงตอนนี้เราคงทราบว่าโครงสร้างของแฟ้มข้อมูลหนึ่ง ๆ มีรายละเอียดอย่างไร เราจึงพร้อมที่จะเขียนโปรแกรมภาษาปาสคาลมาจัดการกับฐานข้อมูลดีเบสได้

แสดงโครงสร้างฐานข้อมูลด้วยปาสคาล

เราจะเริ่มต้นที่การเขียนโปรแกรมอ่านโครงสร้างของฐานข้อมูลดีเบสก่อน เนื่องจากผู้เขียนประสงค์จะเขียนโปรแกรมให้อ่านได้กับทุกแฟ้มข้อมูลของดีเบส จึงกำหนดตัวแปรล่วงหน้าไว้ก่อนดังรูปที่ 3 โดยมีรายละเอียดดังนี้

- DataDbase เป็นตัวแปรมิติเพื่อนำมาเก็บส่วนของข้อมูลในแฟ้มข้อมูลดีเบส ซึ่งตามโปรแกรมนี้จะกำหนดได้ยาว 52000 ไบต์
- NoR สำหรับกำหนดหมายเลขระเบียนข้อมูล
- CountOfRecord เก็บจำนวนของระเบียนข้อมูล
- SizeOfRecord เก็บความยาวของระเบียนข้อมูล

- SizeOfHead เก็บความยาวส่วนหัวของดีเบสหรือส่วนที่ไม่ใช่ข้อมูล
- EditNo สำหรับกำหนดหมายเลขข้อมูลที่จะทำการแก้ไข Head ตัวแปรมิติเพื่อรองรับข้อมูลส่วนหัวของแฟ้มข้อมูลที่อ่านได้
- NameOfField ตัวแปรมิติสำหรับเก็บชื่อเขตข้อมูล กำหนดไว้ 30 เขตข้อมูล
- TypeOfField ตัวแปรมิติสำหรับเก็บชนิดของเขตข้อมูล
- WidthOfField ตัวแปรมิติสำหรับเก็บความกว้างของเขตข้อมูล
- Fdec ตัวแปรมิติสำหรับเก็บจำนวนเลขหลังจุดทศนิยมในกรณีที่ชนิดของเขตข้อมูลเป็นตัวเลข
- CountOfField, Appendck เก็บจำนวนเขตข้อมูลในหนึ่งระเบียนข้อมูล และเงื่อนไขการเพิ่มข้อมูลตามลำดับ
- Dbasefile, DataStr คือตัวแปรเก็บชื่อแฟ้มฐานข้อมูล และวันที่ที่แก้ไขครั้งสุดท้าย

```
PROCEDURE ReadStructure;
var no,s:word; k,ck:byte;

       Procedure dateDbase;
        var st:string[4];
         begin
          DateStr:='';
          str(Head[2],St); DateStr:=Datestr+St+'/';
          str(head[3],St); if head[3]<10 then St:='0'+st;
          DateStr:=Datestr+St+'/';
          str(Head[1],St);  DateStr:=Datestr+'19'+St;
         end;

begin
  assign(f,Dbasefile);  reset(f,1);
  blockread(f,head,10);
  SizeOfHead:=Head[8]+Head[9]*256;
  reset(f,1);
  blockread(f,head,SizeOfHead);
  close(f);
    DatedBASE;
    CountOfRecord:=Head[4]+Head[5]*256;
    SizeOfRecord:=Head[10]+Head[11]*256;
  writeln('SizeOfHead  :',SizeOfHead);
  writeln('====================================');
  writeln;
  writeln('Structure for database : ',DbaseFile);
  writeln('Number of data records : ',CountOfRecord);
  writeln('Date of last update    : ',DateStr);
  writeln('Filed  Field name  Type   Width  Dec');
  CountOfField:=round(SizeOfHead/32)-1;
 for no:=1 to CountOfField  do
  begin
   NameOfField[no]:='          ';
   TypeOfField[no]:=chr(Head[32*no+11]);
   WidthOfField[no]:=Head[32*no+16];
    ck:=0;
      for k:=0 to 9 do
       begin
        if head[32*no+k]=0 then ck:=27;
        if ck<>27 then NameOfField[no][k+1]:=chr(Head[32*no+k]);
       end;
        write(no:5,'  ',NameOfField[no],'    ',TypeOfField[no]);
        write('      ',WidthOfField[no]:5);
      if TypeOfField[no]='N' then
         begin FDec[No]:=Head[32*no+17];
               if Fdec[no]<>0 then write('  ',Fdec[no]:3);
         end;
      writeln;
  end;
   writeln('** Total **                ',SizeOfRecord:5);
   ch:=readkey;
end;
```

รูปที่ 4 ชุดคำสั่งสำหรับแสดงโครงสร้างของแฟ้มข้อมูลด้วยปาสคาล

ส่วนรูปที่ 4 คือส่วนของโปรแกรมที่จะทำการแสดงโครงสร้างของฐานข้อมูล มีส่วนที่จะอธิบายดังนี้ ในชุดคำสั่ง dateDbase คือส่วนที่นำเอาส่วนหัวของแฟ้มฐานข้อมูลมาหาวัน-เดือนปี ซึ่งใช้ Head[2], Head[3], Head [3] เก็บเดือน วัน และปีตามลำดับ มากำหนดให้เป็นตัวแปร DateStr ถ้าประสงค์จะให้แสดงเป็นวันเดือนปีแบบไทย ๆ ก็แก้ไขเองได้

การทำงานเริ่มต้นด้วยการเปิดแฟ้มฐานข้อมูล อ่านส่วนหัวมาเพียง 10 ไบต์ เพื่อหาความยาวของส่วนหัว จาก SizeOfHead:=Head[8]+Head [9]×256; เมื่อทราบส่วนนี้แล้วก็เริ่มต้นอ่านแฟ้มฐานข้อมูลใหม่ด้วยคำสั่ง reset (f,1) อีกครั้ง แต่คราวนี้อ่านส่วนหัวมาเท่ากับค่าของ Size-OfHead ที่คำนวณได้ จากนั้นหาวันที่ที่แก้ไขแฟ้มฐานข้อมูลครั้งสุดท้ายจาก dateDbase; แล้วหาจำนวนของระเบียนข้อมูลจาก CountOfRecord: = Head [4]+Head[5]×256; หาความกว้างของระเบียนข้อมูลจาก SizeOfRecord:= Head[10]+Head[11]×256 ต่อมาจ
แสดงให้ทราบถึงส่วนต่าง ๆ ที่คำนว
ได้บนจอภาพ โดยเราสามารถหาจำนว
ของเขตข้อมูลที่มีในหนึ่งระเบียนข้อมู
ได้จาก CountOfField:=round (Size
OfHead/32)−1;

คราวนี้มาถึงส่วนที่จะแสดงชื่
เขตข้อมูล ชนิดของเขตข้อมูล ความ
กว้างของเขตข้อมูล ซึ่งอยู่ในส่วนขอ
คำสั่งวนรอบ for ของตัวแปร n
ดังนี้ เริ่มด้วยการหาชนิดของเขตข้อมูลจ
TypeOfField[no]:=chr(Head[32×no

1]; หาความกว้างของเขตข้อมูลจาก
VidthOfField[no]:=Head[32×no+
6]; หาชื่อเขตข้อมูลจากส่วนคำสั่ง
นรอบ for ของตัวแปร k ซึ่งมีค่า
ก 0 ถึง 9 คือ 10 ครั้ง โดยหาได้จาก
ameOfField[no] [k+1]:=chr(Head
2×no+k]; ถ้าพบว่าชนิดของเขต
อมูลเป็นตัวเลข (N) ให้แสดงจำนวน
องเลขหลังจุดทศนิยมด้วยคำสั่ง FDec
o]:=Head[32×no+17]; และจบลง
ยแจ้งให้ทราบถึงความยาวของระเบียน
อมูล

เมื่อเรียกใช้ชุดคำสั่งนี้จะได้ส่วน
องโครงสร้างแฟ้มข้อมูล test.dbf
งรูปที่ 5 จะพบว่าเราได้โครงสร้าง
บบเดียวกัน เพียงแต่ ค.ศ.เปลี่ยนจาก
0 มาเป็น 1990 และส่วนบนสุดบอก
ห้ทราบว่าส่วนหัวของแฟ้มนี้มีอยู่ 322
บต์ ซึ่งส่วนนี้ไม่ใช่ส่วนของข้อมูล
รือข้อมูลจะเริ่มตั้งแต่ไบต์ที่ 323 เป็น-
นไป ลองใช้ชุดคำสั่งนี้อ่านแฟ้มฐาน
อมูลอื่น ๆ โดยกำหนดตัวแปร Dbase-
le ใหม่ เราจะพบว่าชุดคำสั่งนี้สามารถ
ช้ได้กับทุก ๆ แฟ้มฐานข้อมูลดีเบส

```
SizeOfHead  :322
====================================

Structure for database : test.dbf
Number of data records : 430
Date of last update    : 11/09/1990
Filed  Field name  Type   Width  Dec
    1  NUMBER       N         3
    2  NAME         C        16
    3  NAME2        C        16
    4  UNIT         C        10
    5  SCHOOL       C        15
    6  POSITION     C         7
    7  LEVEL        C         1
    8  DISTRICT     C        10
    9  PROVINCE     C        10
** Total **                  89
```

รูปที่ 5 แสดงโครงสร้างฐานข้อมูลที่อ่านได้ด้วยปาสคาล

```
PROCEDURE ReadDatabase;
begin
  assign(f,Dbasefile);
  reset(f,1);
  blockread(f,Head,SizeOfHead);
  blockread(f,DataDbase,SizeOfRecord*CountOfRecord);
  close(f);
end;

PROCEDURE DisplayDatabase(R:word);
var no,s,x,xx:word;
begin
   WidthOfField[0]:=0; xx:=1;
   writeln('========== Record no.',R,' ============');
 for No:=1 to CountOfField do
   begin
       write('  ',NameOfField[no],': ');
      if R<=CountOfRecord then
        begin
         x:=(R-1)*SizeOfRecord+xx;
         for s:=x to (x+WidthOfField[No]-1) do write(DataDbase[s]);
        end;
         writeln;   xx:=xx+WidthOfField[no];
   end;
end;
```

รูปที่ 6 ชุดคำสั่งในการอ่านแฟ้มข้อมูลและแสดงข้อมูลหนึ่งระเบียนข้อมูล

ารแสดงข้อมูล

เมื่อเราเข้าใจโครงสร้างของฐาน
อมูลแล้ว ต่อไปผู้เขียนจะขอกล่าวถึง
ารแสดงผลข้อมูลบนจอภาพ ซึ่งเรา
ะต้องทำการอ่านข้อมูลนั้นขึ้นมาก่อน
นที่นี้ผู้เขียนใช้วิธีอ่านข้อมูลทั้งหมด
บไว้ในหน่วยความจำเลย โดยอาศัย
ลักการเดียวกับฐานข้อมูลดีเบสคือ
นไฟล์รหัสแอสกี เราจึงเขียนโปร-
กรมจัดการได้ดังรูปที่ 6 มี 2 ชุดคำสั่ง
งนี้

- ReadDatabase สำหรับอ่านข้อ-
ลเข้ามาในตัวแปร DataDbase

- DisplayDatabase ใช้สำหรับแสดงข้อมูลในระเบียนข้อมูลที่กำหนดให้

จากรูปที่ 6 เริ่มจากการอ่านแฟ้มข้อมูลส่วนหัวเก็บไว้ก่อน จากนั้นค่อยอ่านส่วนของข้อมูลจริงมาใส่ตัวแปร DataDbase โดยให้ข้อมูลมีความยาวเท่ากับ SizeOfRecord × CountOf-Record แล้วปิดแฟ้มเลย เราไม่ต้องคอยกังวลว่าจะออกจากโปรแกรมไปในขณะที่เปิดแฟ้มข้อมูลค้างไว้หรือเปล่า อันอาจก่อให้เกิดปัญหาขึ้นได้เมื่อกลับ

มาใช้แฟ้มข้อมูลนี้ในภายหลังเหมือนกับที่ผู้ใช้ดีเบสเคยประสบกันมาแล้ว

ส่วนชุดคำสั่งการแสดงข้อมูล ได้แก่

- เรียกใช้โดยการกำหนด R คือหมายเลขของระเบียนข้อมูลที่ต้องการ

- กำหนดตัวแปร xx:=1; เพื่อใช้ในการอ่านข้อมูล ซึ่งค่านี้จะเพิ่มตามความกว้างของเขตข้อมูลที่อ่านเริ่มด้วยการเขียนชื่อของเขตข้อมูลด้วยตัวแปร NameOfField[no]

- หาตำแหน่งเริ่มต้นของข้อมูลในเขตข้อมูลนั้น ๆ จาก x:=(R−1)×SizeOfRecord+xx; จากนั้นจึงแสดงข้อมูลที่เก็บไว้ในเขตข้อมูลที่ได้จากการวนรอบ for ของตัวแปร s โดยเริ่มค่า s ตั้งแต่ x ถึง x+WidthOfField[no]−1 ทำให้สามารถแสดงข้อมูลที่เก็บไว้นั้นได้ด้วยตัวแปร DataDbase[s]

- เพิ่มค่าตัวแปร xx เป็น xx:=xx+WidthOfField[No]; แล้วแสดงข้อมูลของเขตข้อมูลถัดไปจนกว่าการวนรอบ for ของตัวแปร no จะสิ้นสุดเป็นการจบส่วนของการแสดงข้อมูล

```
PROCEDURE AppendDatabase;
var no,s,x,xx,R:word;  row,k:byte;  InputStr:string;
begin
   clrscr;      writeln(' Append Database ');
   R:=CountOfRecord+1;  Head[4]:=R mod 256; Head[5]:=trunc(R/256);
   DisplayDatabase(R);  WidthOfField[0]:=0;
   xx:=1;  row:=3;      CountOfRecord:=R;
 for No:=1 to CountOfField do
   begin
      repeat
        gotoxy(15,row);   writeln('                                    ');
        gotoxy(15,row);   readln(InputStr);
      until length(InputStr)<=WidthOfField[no];
     if Inputstr<>'' then
      begin
        x:=(R-1)*SizeOfRecord+xx;  k:=1;
        for s:=x to (x+WidthOfField[No]-1) do
         begin
           if k<=length(Inputstr) then DataDbase[s]:=InputStr[k]
                                  Else DataDbase[s]:=#32;
           inc(k);
         end;
      end;
        xx:=xx+WidthOfField[no];  inc(row);
   end;
end;

PROCEDURE SaveDatabase(Filename:string);
var year,Month,day,minute:word;
begin
  getdate(year,Month,day,minute);
  Head[1]:=year-1900; Head[2]:=Month; Head[3]:=day;
  assign(f,filename);
  rewrite(f); reset(f,1);
  blockwrite(f,Head,SizeOfHead);
  blockwrite(f,DataDbase,SizeOfRecord*CountOfRecord);
  ch:=#26; Blockwrite(f,ch,1);
  close(f);
end;
```

รูปที่ 7 ชุดคำสั่งในการเพิ่มข้อมูลหนึ่งระเบียนข้อมูล

เพิ่มข้อมูลในระเบียนข้อมูล

เมื่อเราประสงค์จะเพิ่มข้อมูลในฐานข้อมูลก็ทำได้ เช่น เมื่ออยู่ในดีเบสจะใช้คำสั่ง appe เพื่อเพิ่มข้อมูลตามความต้องการของเรา เมื่อเราใช้ปาสคาลจัดการแทนก็ทำแบบเดียวกัน คือต้องเขียนชุดคำสั่งขึ้นมาจัดการ ถ้าเขียนเสร็จแล้วก็เรียกใช้ได้ทันทีแบบเดียวกับที่เราเรียกใช้ในดีเบส

ชุดคำสั่ง AppendDatabase จะเป็นส่วนของการเพิ่มข้อมูลในแฟ้มฐานข้อมูล ขอให้ดูโปรแกรมในรูปที่ 7 ก่อน ชุดคำสั่ง SaveDatabase เป็นชุดคำสั่งในการเก็บแฟ้มข้อมูลที่ได้เพิ่มข้อมูลแล้ว

คราวนี้มาดูคำสั่งที่ใช้ในชุดคำสั่ง AppendDatabase กันบ้าง ซึ่งพอจะอธิบายได้ดังนี้

- กำหนดค่าตัวแปร R:=CountOfRecord+1 แล้วทำการกำหนดส่วนหัวของแฟ้มฐานข้อมูลในส่วนที่เก็บจำนวนระเบียนข้อมูลคือออฟเซตที่ 4-5 เสียใหม่ เพื่อให้จำนวนระเบียนข้อมูลเพิ่มขึ้นอีก 1 ระเบียนโดยใช้คำสั่ง Head[4]:=R mod 256; Head[5]:=trunc (R/256);

- แสดงรายชื่อเขตข้อมูลบนจอภาพจาก DisplayDatabase(R);

- กำหนดตัวแปรใหม่ดังนี้ xx:=1; CountOfRecord:=R;

- ในช่วงวนรอบ for ของตัวแปร no เราต้องป้อนข้อมูลในตัวแปรชื่อ Inputstr โดยกำหนดว่าความยาวของตัวแปร Inputstr และต้องไม่มากเกินกว่าความกว้างของเขตข้อมูลนั้น คือไม่เกินตัวแปร WidthOfField[n

ถ้าการป้อนถูกต้องก็นำข้อมูลม ใส่ลงในเขตข้อมูลที่เราใช้งานอยู่ (No) โดยกำหนดจุดเริ่มต้นของเข ข้อมูลด้วยค่า x:=(R-1)×SizeO Record+xx; แล้วแทนที่ตัวแปร Dat base[s] ให้เป็นข้อมูลตามการป้อนน โดยดูได้จากช่วงของการวนรอบ f ของตัวแปร s หลังจากจบการใส่ข้อมู ในเขตข้อมูลแรกให้เพิ่มค่าตัวแปร x ใหม่ตามความกว้างของเขตข้อมูล โ ดูจากคำสั่ง xx:=xx+WidthOfFie

no]; และตามตัวอย่างนี้จะแก้ไขเขต
ข้อมูลได้ไม่เกินค่า row=24

ส่วนการเก็บแฟ้มข้อมูลจะเป็น
ไปตามคำสั่งในชุดคำสั่ง SaveDatabase
ซึ่งเราคงทำความเข้าใจได้ไม่ยาก และ
สาเหตุที่ต้องกำหนดชื่อแฟ้มมาด้วยนั้น
ก็เพราะบางคนอาจเก็บลงในชื่อใหม่
หากประสงค์จะให้ดำเนินไปตามแนว-
ทางของดีเบสควรตัดพารามิเตอร์ file-
name ทิ้งไป และเปลี่ยนการกำหนด
ชื่อแฟ้มจากของเดิมให้เป็น DbaseFile
จะเป็นการเก็บลงในชื่อเดิม การเก็บนี้
ต้องเพิ่มการเก็บวันที่ที่แก้ไขครั้งสุดท้าย
ด้วย คือแก้ไขที่ Head[1]–Head[3]
โดยการอ่านวันเดือนปีด้วยคำสั่ง get-
time (year, month, day, minute);
และแก้ไขตัวแปรของแฟ้มข้อมูลใน
ส่วนที่เก็บปีเดือนวันใหม่ดังนี้ Head
[1]:=year–1900; Head[2]:month;
Head[3]:=day; ทำให้การเก็บแฟ้ม
ข้อมูลเป็นไปอย่างสมบูรณ์ตามแนวคิด
ของดีเบสทุกประการ

โปรแกรมหลัก

เมื่อเราเขียนชุดคำสั่งเรียบร้อย
ลำดับต่อมาเราต้องเขียนส่วนการจัดการ
ตามที่เราต้องการดังรูปที่ 8 มีรายละเอียด
ดังนี้

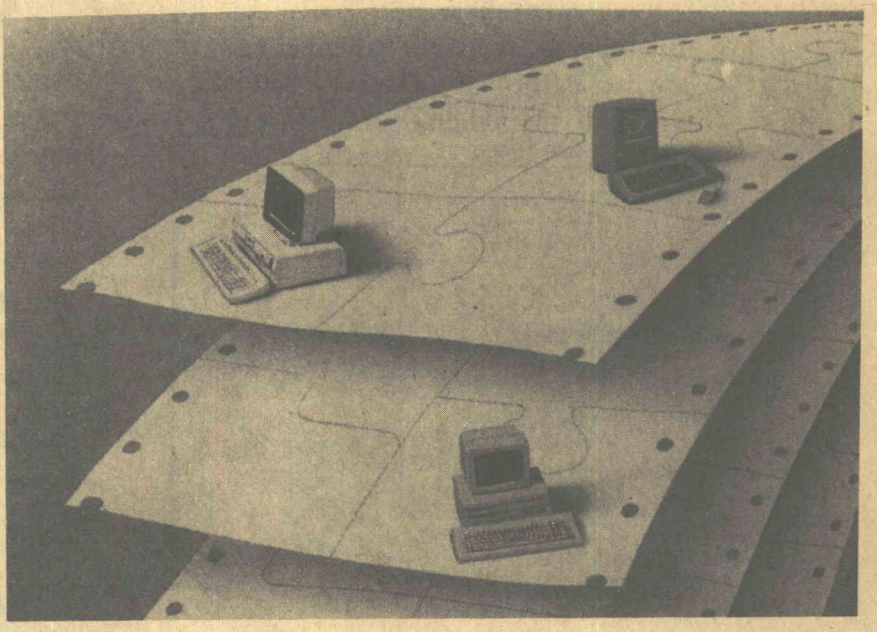
- กำหนดชื่อแฟ้มฐานข้อมูล
ดีเบสที่จะอ่าน จากตัวอย่างคือ Dbase-
File:='test.dbf';

- ทำการอ่านโครงสร้างและ
ข้อมูลจากคำสั่ง ReadStructure; Read-
Database; และกำหนดให้ตัวแปร NoR:
=1; เพื่อเริ่มแสดงข้อมูลตั้งแต่ระเบียน
ข้อมูลที่ 1 ไปจนถึงระเบียนสุดท้าย
หรือกดแป้น Esc ที่ปรากฏในช่วงของ
การวนรอบ repeat

```
{=================== Read dBase_files with pascal ================}

begin
  clrscr;  Dbasefile:='test.dbf';
  ReadStructure;   ReadDatabase;  NoR:=1;
   repeat
       clrscr;
       DisplayDatabase(NoR);  inc(NoR);
       ch:=readkey; if ch=#27 then NoR:=CountOfRecord+1;
   until NoR>CountOfRecord;
     Appendck:=0;
   if Appendck<>0 then  begin
     AppendDatabase;  DisplayDatabase(CountOfRecord);
     SaveDatabase('test.dbf');
                    end;
     ch:=readkey;
end.
```

รูปที่ 8 ส่วนของโปรแกรมหลักเพื่อจัดการฐานข้อมูลด้วยปาสคาล



ตัวโปรแกรมนี้จะไม่มีการเพิ่ม
ข้อมูลใด ๆ หากเราต้องการเพิ่มข้อมูล
ให้เปลี่ยนจากตัวแปร Appendck ไปใช้
ค่าอื่นที่ไม่ใช่ 0 หากค่าตัวแปร Ap-
pendck ไม่เป็น 0 จะมีการเพิ่มข้อมูล
และทำการเก็บแฟ้มข้อมูลในชื่อเดิม
เนื่องจากกำหนดไว้ว่า SaveDatabase
('test.dbf');

ก่อนจากกัน

คงจะต้องจบตอนที่หนึ่งไว้เพียง
เท่านี้ หากผู้อ่านมีสิ่งใดที่อยากให้เขียน
ถึง กรุณาแจ้งให้ทางกองบรรณาธิการ
วารสารไมโครฯทราบด้วย สำหรับตอน
ที่สองนั้นผู้เขียนจะพูดถึงการแก้ไข
ข้อมูล การค้นหาข้อมูล การเรียงลำดับ
ข้อมูล ฯลฯ ถ้าสนใจคงต้องอดใจรอ
ตอนที่สองในฉบับหน้า